Proceedings of CONF-MLA 2026 Symposium: Intelligent Systems and Automation: AI Models, IoT, and Robotic Algorithms
DOI: 10.54254/2755-2721/2026.LD37426

Application of Graph Neural Networks in Computational Fluid Mechanics

Anyi Xiao

Gaston Day School Shanghai Shangde, Shanghai, China
xiaoanyi19@163.com

Abstract. Computational fluid dynamics (CFD) discretizes the Navier-Stokes equations on computational meshes to obtain flow-field quantities like velocity, pressure, and density through numerical iteration, but traditional solvers, though reliable in accuracy, are computationally expensive and time-consuming for high-Reynolds-number turbulence and complex unsteady flows. With geometric deep learning, graph neural networks (GNNs) overcome the limitation of conventional neural network (CNNs) that only handle grid-structured Euclidean data; by message-passing, they naturally fit unstructured CFD meshes and capture spatial correlations. This paper systematically reviews GNN fundamentals and core fluid mechanics, including turbulence characteristics and mesh type classifications, and then focuses on typical GNN applications in flow-field prediction, turbulence modeling, and complex-flow simulation. Existing challenges such as high training costs, insufficient generalization, and limited computational efficiency are analyzed, and promising future research directions are discussed. The review indicates GNNs are efficient surrogates for traditional solvers, offering a pathway to intelligent, fast, high-fidelity simulation.

Keywords: Graph neural network, computational fluid dynamics, turbulence modeling, flow-field prediction, mesh-based simulation

1. Introduction

Fluid mechanics is a fundamental discipline studying fluid motion and fluid-solid interactions, underpinning aerospace, mechanical, civil, and ocean engineering. Since analytical solutions are available only for a few idealized cases, numerical simulation has become the primary approach for practical flows [1]. Mature methods like finite volume and finite element have been developed and can deliver reliable results under various conditions. However, traditional CFD solvers show clear limitations as industrial demands become stricter.

In recent years, artificial intelligence has been increasingly integrated into fluid mechanics. Data-driven deep learning methods can mine hidden physical laws from large volumes of flow-field data and achieve rapid prediction and simulation [2, 3]. Early attempts, such as convolutional neural networks for steady flow approximation, demonstrated that a network can estimate a velocity field orders of magnitude faster than a GPU-accelerated CFD solver while maintaining low error [4]. Nevertheless, conventional deep learning architectures like CNNs and RNNs are designed for regular, structured data and are difficult to apply directly to irregular meshes commonly used in

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.LD37426

CFD. By contrast, graph neural networks (GNNs) take graph-structured data as their object and are naturally compatible with unstructured meshes [5, 6]. A CFD mesh can be abstracted to a graph straightforwardly: mesh nodes as graph nodes, and their adjacency relationships as edges. Through neighbor feature aggregation and iterative updates, GNNs can model spatial interaction and temporal evolution of fluid flows [7].

This paper systematically reviews basic theories of GNNs and fluid mechanics, highlights typical applications of GNNs in computational fluid mechanics, analyzes key difficulties restricting current technology, and proposes targeted future research directions.

2. Overview of graph neural networks

2.1. Basic concepts and the message-passing mechanism

A graph neural network is a class of geometric deep learning models specifically designed for graph-structured data [2, 6]. A complete graph can be defined as G = (V, E), where V denotes the set of nodes and E denotes the set of edges connecting the nodes. In graph data, each node carries its own feature attributes, while each edge records the topological connection and, optionally, spatial attributes between two nodes. Different from Euclidean data such as images and texts, which have fixed local neighborhoods, graph data have irregular neighborhood structures, variable numbers of neighbors and flexible topologies. These properties match the discretized mesh structure of CFD naturally.

The core design concept of GNNs is the message-passing mechanism, which simulates the information exchange between adjacent nodes in the graph. The whole process can be divided into three steps: gathering, aggregation and update. First, each target node gathers the features of all its neighboring nodes together with the associated edge attributes. Second, an aggregation function, such as mean, sum or maximum, fuses the collected neighbor information into an effective message vector. Third, the fused message is combined with the original feature of the target node to update the node state. By stacking multiple GNN layers, the model can gradually capture both short-range and long-range spatial correlations and achieve global modeling of the entire graph. This learning paradigm is analogous to that of CNNs in image processing, with one crucial difference: whereas a CNN operates on a fixed rectangular neighborhood, a GNN operates on a neighborhood whose size and structure vary from node to node.

2.2. Mainstream GNN models

Along with the development of graph deep learning, a variety of classic GNN models have been derived, among which the graph convolutional network (GCN), the graph attention network (GAT) and the message passing neural network (MPNN) are the most widely used in fluid simulation research.

The GCN is the earliest and most fundamental graph convolution model [1]. It realizes the graph convolution operation on the basis of spectral graph theory and spatial neighbor aggregation, and simplifies the complex Laplacian matrix operation to complete node feature extraction efficiently. GCN has fast calculation speed and stable performance, which makes it suitable for regular and small-scale fluid mesh modeling.

The GAT introduces the attention mechanism into the graph convolution framework [1]. Instead of assigning equal weights to all neighbors, it dynamically computes the importance weight of each neighbor through self-attention. This allows the model to distinguish the influence of different mesh

DOI: 10.54254/2755-2721/2026.LD37426

nodes on the flow field, which is beneficial for complex turbulent flows with uneven spatial distributions. The MPNN is a general graph message-passing framework that covers almost all mainstream GNN variants [2]. It standardizes the message transmission and node update rules and is highly flexible. Most mesh-based fluid simulation GNN models are constructed on the basis of the MPNN framework [3].

2.3. Core task types of GNNs

According to the prediction objective, GNN tasks can be divided into three categories: node-level tasks, edge-level tasks and graph-level tasks. Node-level tasks predict the attribute value of each single node and are the most common tasks in fluid mechanics, corresponding to the prediction of flow-field parameters at each mesh node. Edge-level tasks predict the connection relationship and attribute characteristics between nodes, which can reflect the interaction and transmission between fluid mesh cells. Graph-level tasks integrate all node features through a graph pooling operation to obtain the global state of the whole flow field, which is suitable for the overall performance evaluation of a fluid simulation. In practice, a single GNN model can also be trained in a multi-task manner so that node-level outputs and graph-level outputs are produced simultaneously, which improves the data efficiency of the training process.

3. Core theoretical knowledge of fluid mechanics

3.1. Characteristics of turbulent flow

Fluid flow is generally divided into two basic states: laminar flow and turbulent flow. The flow state is mainly determined by the Reynolds number (Re). Laminar flow is a stable, layered state in which streamlines are regular and mixing between fluid layers is negligible. When the Reynolds number exceeds a critical value, the flow undergoes transition from laminar to turbulent flow, which is the most common flow state in industrial applications.

Turbulent flow possesses prominent multi-scale and unsteady characteristics, and its core feature is the energy cascade phenomenon [4]. In a turbulent flow field, large-scale vortex structures are continuously generated under the combined action of boundary disturbances and fluid inertia. As the flow develops, large vortices break into medium-scale vortices, and these medium-scale vortices further decompose into tiny eddies. Eventually, the energy of the small eddies is dissipated as viscous heat. This multi-scale evolution gives turbulent flow strong spatial and temporal randomness.

3.2. Classification of CFD mesh types

Mesh discretization is the premise of CFD numerical simulation, and the type and quality of the mesh directly determine the simulation accuracy and computational efficiency. CFD meshes are mainly divided into structured meshes and unstructured meshes, which differ significantly in structural characteristics and applicable scenarios.

A structured mesh is a regular grid with uniform arrangement and fixed topological relations. The mesh nodes are arranged in an orderly manner, and each interior node has the same number of adjacent nodes. Structured meshes usually yield high calculation accuracy, small numerical dissipation and fast iteration speed, and they are suitable for regular geometric models such as flat-

DOI: 10.54254/2755-2721/2026.LD37426

plate flow and circular-pipe flow. However, their geometric adaptability is poor, and it is difficult for them to fit complex curved boundaries and irregular geometries.

An unstructured mesh has no fixed arrangement rule or topological constraint, and the number of adjacent nodes of each mesh cell is variable. It can flexibly fit complex geometric boundaries such as airfoils, vehicles and mechanical structures and therefore has strong geometric adaptability. It is the mainstream mesh type for industrial complex-flow simulation. However, the irregular topology of unstructured meshes prevents traditional CNN models from extracting spatial features effectively, which creates the core application scenario for GNNs.

4. Applications of GNN in computational fluid mechanics

4.1. Flow-field prediction

Flow-field prediction is the most fundamental and widely used application of GNNs in fluid mechanics. Its core goal is to build a surrogate model of traditional CFD solvers that can quickly predict the steady and unsteady flow-field distribution under given geometric boundaries and initial conditions, thereby greatly shortening the simulation cycle.

The mainstream technical framework for GNN-based flow-field prediction is the Encoder-Processor-Decoder structure [3]. The encoder maps the CFD mesh graph into a high-dimensional feature space. The processor performs multi-layer graph message passing to capture the spatial interaction rules of fluid motion and the temporal evolution of the flow field. The decoder restores the high-dimensional features into the actual flow-field parameters. A representative implementation of this framework is MeshGraphNet, which has been shown to generalize across different mesh sizes and physical systems, including incompressible fluids, cloth and rigid bodies, and to support long-horizon rollouts of unsteady dynamics.

In steady flow prediction, GNNs can quickly predict the steady pressure field and velocity field of classic cases such as flow past a cylinder and flow around an airfoil, with an accuracy close to that of traditional CFD simulations. In unsteady transient flow prediction, GNNs realize the iterative prediction of time-series flow fields and simulate dynamic phenomena such as vortex shedding and flow-field pulsation, which avoids the long iteration time of traditional transient CFD simulations. In the training stage, a GNN surrogate is trained on a dataset of paired inputs and outputs collected from a number of offline CFD runs; once trained, it can generalize to unseen initial conditions and geometries within the training distribution. Compared with traditional numerical methods, GNN-based flow-field prediction can reduce the calculation time from hours or days to seconds, providing strong technical support for rapid geometric optimization and working-condition screening in industrial design.

4.2. Turbulence modeling

Turbulence modeling is the core difficulty of fluid simulation. Traditional Reynolds-averaged turbulence models show large errors in the simulation of complex vortex flows, whereas high-precision methods such as large-eddy simulation and direct numerical simulation consume huge computational resources. GNNs provide a new data-driven method for turbulence modeling.

GNN-based turbulence modeling takes discrete turbulent flow-field meshes as the research object and learns the nonlinear mapping between the mean flow-field parameters and the turbulent fluctuation characteristics through the graph message-passing mechanism, thereby constructing an intelligent turbulence closure model [5]. Different from traditional empirical turbulence models that

DOI: 10.54254/2755-2721/2026.LD37426

rely on manually designed formulas, GNNs can automatically mine the multi-scale vortex evolution law and the energy dissipation characteristics of turbulence from massive high-fidelity flow-field data. The required supervision signal is usually obtained by projecting high-fidelity simulation data, such as the results of direct numerical simulation, onto the corresponding coarse meshes. This method can correct the prediction errors of traditional turbulence models in complex conditions such as flow separation and high-speed turbulent jets, improve the simulation accuracy of unsteady turbulent flows, and significantly reduce the computational cost of high-fidelity turbulence simulation.

4.3. Complex-flow simulation

Traditional CFD simulation often fails or has low accuracy when facing complex flows such as multi-scale flows, separated boundary flows and flows around irregular geometries. Relying on the powerful topological adaptability of unstructured meshes, GNNs show unique advantages in complex-flow simulation.

GNNs can adapt to the irregular mesh distribution of complex geometric models, accurately capture the sudden flow-field changes at complex boundaries, and simulate complex phenomena such as boundary-layer separation, vortex merging and flow reattachment. At the same time, multi-layer graph stacking enables the model to capture both the small-scale viscous flow characteristics near the wall and the large-scale mainstream flow evolution, thus achieving multi-scale unified modeling of complex flows. Because the graph representation is not tied to any particular mesh topology, a well-trained GNN can also be evaluated on meshes of different densities and connectivities, which makes it more flexible than models that require a fixed computational grid. At present, GNNs have been successfully applied to aerospace airfoil flows, internal machinery flows and ocean flows, showing promising engineering application prospects.

5. Discussion

5.1. Existing challenges

Although GNNs have made considerable progress in fluid simulation, they are still restricted by data quality, model performance and computing resources, and several challenges limit their large-scale industrial application.

First, the training cost of the model is extremely high. High-fidelity turbulent flow-field data mainly come from expensive CFD numerical simulations and experimental measurements, whose acquisition cycle is long and resource consumption is huge, resulting in the scarcity of high-quality labeled fluid data. Moreover, the unstructured meshes of complex flows contain a large number of nodes and edges, forming super-large-scale graph networks. Second, the generalization ability of the model is insufficient. Most current GNN fluid simulation models perform well on the training conditions and on similar geometric structures, but their prediction accuracy drops sharply when they face new geometric shapes, different Reynolds-number conditions and unknown boundary conditions. Because such failures are difficult to anticipate without running a full numerical solver as a reference, the reliability of data-driven surrogates in safety-critical engineering remains an open question. Third, the overall computational efficiency still needs to be improved. Although the inference speed of GNNs is far faster than that of traditional CFD iteration, the message aggregation and matrix operations of large-scale graphs still have high computational overhead. Compared with

DOI: 10.54254/2755-2721/2026.LD37426

lightweight CNN models, GNNs have an obvious speed disadvantage in ultra-high-resolution flow-field prediction, which limits their application to high-fidelity, full-scale fluid simulation.

5.2. Future research directions

To address the above challenges, future research on GNNs in computational fluid mechanics will focus on physical constraint optimization, generalization improvement and computational efficiency upgrading.

The first direction is to develop physics-informed graph neural networks. By embedding the Navier-Stokes equations and the conservation laws of mass and momentum into the loss function and the message-passing rules of the model, the outputs can be constrained to conform to physical reality, which eliminates non-physical prediction errors and improves the physical interpretability and reliability of the model [8]. The second direction is to improve the cross-condition generalization ability of the model. Meta-learning and transfer-learning algorithms can be introduced to realize fast adaptation of the model to new geometries and new Reynolds-number conditions with only a few samples. The third direction is to optimize the graph computing framework and improve simulation efficiency. Technologies such as sparse graph convolution, graph pruning and hierarchical graph modeling can reduce the redundant computation of large-scale graph networks, lower memory occupation and computational overhead, and realize efficient simulation of ultra-high-resolution complex flow fields.

6. Conclusion

This paper has systematically studied the application of graph neural networks in computational fluid mechanics. It has expounded the basic principles and core task modes of GNNs, sorted out the key theoretical characteristics of fluid turbulence and the classification of CFD mesh types, and analyzed the application value of GNNs in fast flow-field prediction, intelligent turbulence modeling and complex-flow simulation.

Different from traditional deep learning models that are limited to structured data, GNNs adapt to the topology of unstructured CFD meshes through the unique message-passing mechanism, break through the efficiency bottleneck of traditional CFD numerical simulation, and realize fast and intelligent prediction of fluid flow fields. The related research results demonstrate that GNNs have excellent performance in accelerating fluid simulation and optimizing turbulence modeling.

At present, the application of GNNs in fluid mechanics still faces many problems, such as high training cost, insufficient generalization and limited computational efficiency. It should be emphasized that GNNs are not intended to replace the well-established numerical methods entirely; rather, they are best viewed as complementary tools that accelerate specific stages of the simulation pipeline. In the future, with the continuous integration of physical constraint algorithms, few-shot learning and graph computing optimization technologies, physics-consistent, highly generalizable and efficient GNN fluid simulation models will be developed. The deep integration of graph neural networks and computational fluid mechanics will open up a new direction for intelligent fluid simulation and has broad value in both academic research and industrial application.

References

[1] Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 1–14. Toulon.

DOI: 10.54254/2755-2721/2026.LD37426

[2] Battaglia, P. W., Hamrick, J. B., Bapst, V., et al. (2018). Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*.
[3] Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A., et al. (2020). Learning mesh-based simulation with graph networks. *arXiv preprint arXiv:2010.03409*.
[4] Pope, S. B. (2000). *Turbulent flows*. Cambridge University Press.
[5] Duraisamy, K., Iaccarino, G., & Xiao, H. (2019). Turbulence modeling in the age of data. *Annual Review of Fluid Mechanics, 51*, 357–377.
[6] Bronstein, M. M., Bruna, J., LeCun, Y., et al. (2017). Geometric deep learning: Going beyond Euclidean data. *IEEE Signal Processing Magazine, 34*(4), 18–42.
[7] Guo, X., Li, W., & Iorio, F. (2016). Convolutional neural networks for steady flow approximation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 481–490). ACM. New York.
[8] Kochkov, D., Smith, J. A., Alieva, A., et al. (2021). Machine learning-accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences, 118*(21), e2101784118.